2.5.6 Skolemizace. Poznamenejme, že termíny „skolemizace“, „skolemizační konstanta“ a „skolemizační funkční symbol“ jsou odvozeny od jména norského matematika — logika Thoralfa Skolema.

Skolemizací nahradíme sentenci ψ sentencí ψ' takovou, že sentence ψ je splnitelná právě tehdy, když je splnitelná sentence ψ' (obecně ale může existovat interpretace, ve které je ψ pravdivá, ale ψ' nikoli). Dříve než ukážeme obecný postup, uvedeme čtyři příklady.

2.5.7 Příklad 1. Najděme klauzuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \exists x\, P(x)$.

Hledanou sentencí je $\psi' = P(a)$, kde a je nějaký nový konstantní symbol. Není obtížné nahlédnout, že formule ψ je tautologickým důsledkem formule ψ'. Navíc, má-li ψ model, můžeme interpretovat konstantu a tak, abychom dostali model ψ'.

Konstantní symbol a použitý v minulém příkladě, se nazývá *skolemizační konstanta*.

2.5.8 Příklad 2. Najděme klauzuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \exists x\, \exists y\, Q(x, y)$.

Nejprve skolemizujeme první existenční kvantifikátor, tj. $\exists x$. Tím dostaneme sentenci je $\exists y\, Q(a, y)$, kde a je nějaký nový konstantní symbol. Nyní skolemizujeme druhý existenční kvantifikátor $\forall y$ a dostáváme klauzuli $\psi' = Q(a, b)$. Opět musí platit, že b je nový konstantní symbol, který se nikde jinde neobjevuje.

Opět není obtížné nahlédnout, že sentence ψ je tautologickým důsledkem sentence ψ'. Navíc, má-li ψ model, můžeme interpretovat konstanty a, b tak, abychom dostali model ψ'.

2.5.9 Příklad 3. Najděme klauzuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \forall x\, \exists y\, Q(x, y)$.

Formuli ψ nahradíme sentencí $\psi' = \forall x Q(x, f(x))$, kde f je nový unární funkční symbol. Nyní již opět platí, že sentence ψ je splnitelná právě tehdy, když je splnitelná sentence ψ'.

Funkčnímu symbolu f se říká *skolemizační funkční symbol.*

2.5.10 Příklad 4. Najděme klauzuli ψ', která je splnitelná právě tehdy, když je splnitelná sentence $\psi = \forall x\, \exists y\, \forall z\, \exists u\, R(x, y, z, u)$.

Skolemizovat začneme kvantifikátory od kořene syntaktického stromu ψ; tedy nejprve odstraníme $\exists y$. Protože před tímto kvantifikátorem je jeden obecný kvantifikátor $\forall x$, nahradíme $\exists y$ novým unárním funkčním symbolem $f(x)$. Dostáváme tedy sentenci $\forall x\, \forall z\, \exists v\, R(x, f(x), x, v$. Nyní při skolemizaci $\exists v$ od tohoto kvantifikátoru do kořene syntaktického stromu máme dva universální kvantifikátory; ano, $\forall x$ a $\forall z$. Zavedeme tedy nový binární funkční symbol $g(x, z)$. Výsledná sentence ψ' je rovna $\psi' = \forall x\, \forall z\, R(x, f(x), z, g(x, z))$, kde f je nový unární funkční symbol a g je nový binární funkční symbol. Nyní opět platí, že sentence ψ je splnitelná právě tehdy, když je splnitelná formule ψ'.

2.5.11 Obecný postup (bod 5 z postupu 2.5.4).

5. Obsahuje-li formule existenční kvantifikátor, nahradíme každou uzavřenou podformuli tvaru $\forall x_1 \dots \forall x_n\, \exists y\, \alpha(x_1, \dots, x_n, y)$ formulí $\forall x_1 \dots \forall x_n\, \alpha(x_1, \dots, x_n, f(x_1, \dots, x_n))$, kde f je libovolný *nový* funkční symbol arity n. Je-li $n = 0$, použijeme *nový* konstantní symbol a. Tomuto procesu se říká *skolemizace*, funkčnímu symbolu f *skolemizační funkční symbol*, konstantě a *skolemizační konstanta.* Pokračujeme podle kroku 6 z 2.5.4.

Uvědomte si, že proměnné $x_1, \dots, x_n$ jsou právě všechny proměnné vázané obecným kvantifikátorem, na které narazíme při postupu syntaktickým stromem od $\exists y$ směrem ke kořeni.

2.5.12 Resolventy klauzulí. Ve výrokové logice jsme resolventy vytvářeli tak, že jsme si vždy vzali dvě klauzule, které obsahovaly dvojici komplementárních literálů, a výsledná resolventa byla disjunkcí všech ostatních literálů z obou klauzulí. Situace v predikátové logice je složitější. Dříve než zadefinujeme pojem resolventy dvou klauzulí v predikátové logice, uvedeme postup vytváření resolvent na příkladech.

2.5.13 Příklad 1. Najděme resolventu klauzulí

$$K_1 = \forall x \, \forall y \, (P(x) \vee \neg Q(x,y)) \quad \text{a} \quad K_2 = \forall x \, \forall y \, (Q(x,y) \vee R(y)),$$

kde P a R jsou unární predikátové symboly a Q je binární predikátový symbol, x, y jsou proměnné.

Klauzule K_1 a K_2 obsahují dvojici komplementárních literálů, totiž $\neg Q(x,y)$ je literál K_1 a $Q(x,y)$ je literál K_2. Obdobně jako ve výrokové logice resolventou klauzulí K_1 a K_2 je klauzule $K = \forall x \, \forall y \, (P(x) \vee R(y))$. (Uvědomte si, že jsme v klauzulích K_1 a K_2 vynechali dvojici komplementárních literálů a klauzule K se skládá ze zbylých literálů.)

2.5.14 Příklad 2. Najděme resolventu klauzulí

$$K_1 = \forall x \, (P(x) \vee \neg Q(x)) \quad \text{a} \quad K_2 = Q(a) \vee R(b),$$

kde P, Q jsou unární predikátové symboly a a, b jsou konstanty.

Klauzule K_1 a K_2 neobsahují dvojici komplementárních literálů, neboť negace literálu $Q(a)$ není literál $\neg Q(x)$ a naopak. Přitom však literál $\neg Q(x)$ odpovídá formuli $\forall x \, \neg Q(x)$, a tedy zahrnuje i $\neg Q(a)$. Proto při substituci konstanty a za proměnnou x dostáváme klauzule

$$K_1' = P(a) \vee \neg Q(a) \quad \text{a} \quad K_2' = Q(a) \vee R(b)$$

a jejich resolventa je klauzule $P(a) \vee R(b)$. (Uvědomte si, že platí $K_1 \models K_1'$.)

2.5.15 Příklad 3. Najděme resolventu klauzulí

$$K_1 = \forall x \, \forall y \, (\neg P(x,y) \vee Q(x,y,a)) \quad \text{a} \quad K_2 = \forall z \, \forall v \, (\neg Q(g(v),z,z) \vee R(v,z)),$$

kde P, R jsou binární predikátové symboly, Q je predikátový symbol arity 3 a a je konstanta.

Pokusíme se vhodnou substitucí vytvořit z $Q(x,y,a)$ a $\neg Q(g(v),z,z)$ dvojici komplementárních literálů. Toho dosáhneme substitucí: za proměnné y a z dosadíme konstantu a, a za proměnnou x dosadíme term $g(v)$. Tím dostaneme komplementární literály

$$Q(g(v),a,a) \qquad \neg Q(g(v),a,a).$$

Provedením substituce na celé klauzule, dostaneme klauzule K_1' a K_2', jejichž resolventa bude i resolventou klauzulí K_1, K_2. Tedy $K_1' = \forall v \, \neg(P(g(v),a) \vee Q(g(v),a,a))$, $K_2' = \forall v \, (\neg Q(g(v),a,a) \vee R(v,a))$ a hledaná resolventa je $K = \forall v \, (\neg P(g(v),a) \vee R(v,a))$. (Uvědomte si, že platí $K_1 \models K_1'$ a $K_2 \models K_2'$.)

2.5.16 Poznámka. Ne vždy resolventa existuje. Resolventa klauzulí

$$K_1 = \forall x \, (\neg P(x) \vee Q(f(x),a)) \quad \text{a} \quad K_2 = \forall y \, \forall z \, (\neg Q(y,y) \vee R(f(y),z)),$$

neexistuje (zde P je unární predikátový symbol, Q a R jsou binární predikátové symboly, f je unární funkční symbol a a je konstanta). Případy, kdy existuje či neexistuje vhodná substituce, řeší unifikační algoritmus. V případě, že substituce existuje, unifikační algoritmus najde nejobecnější substituci též nazývanou *maximální unifikátor*. Zhruba řečeno, maximální unifikátor získáme tak, 6e provedeme pouze nutné substituce a žádné jiné.

2.5.17 Unifikační algoritmus.

Vstup: Dva pozitivní literály L_1, L_2, které nemají společné proměnné.
Výstup: Hlášení `neexistuje` v případě, že hledaná substituce neexistuje,
v opačném případě substituce ve tvaru množiny prvků tvaru x/t,
kde x je proměnná, za kterou se dosazuje,
a t je term, který se za proměnnou x dosazuje.

1. Položme $E_1 := L_1$, $E_2 := L_2$, $\theta := \emptyset$.
2. Jsou-li E_1, E_2 prázdné řetězce, stop. Množina θ určuje hledanou substituci. V opačném případě položíme $E_1 := E_1\theta$, $E_2 := E_2\theta$ (tj. na E_1, E_2 provedeme substituci θ).
3. Označíme X první symbol řetězce E_1, Y první symbol řetězce E_2.
4. Je-li $X = Y$, odstraníme X a Y z počátku E_1 a E_2. Jsou-li X a Y predikátové nebo funkční symboly, odstraníme i jim příslušné závorky a jdeme na krok 2.
5. Je-li X proměnná, neděláme nic.
   Je-li Y proměnná (a X nikoli), přehodíme E_1, E_2 a X, Y.
   Není-li ani X ani Y proměnná, stop. Výstup `neexistuje`.
6. Je-li Y proměnná nebo konstanta, položíme $\theta := \theta \cup \{X/Y\}$. Odstraníme X a Y ze začátků řetězců E_1 a E_2 (spolu s čárkami, je-li třeba) a jdeme na krok 2.
7. Je-li Y funkční symbol, označíme Z výraz skládající se z Y a všech jeho argumentů (včetně závorek a čárek). Jestliže Z obsahuje X, stop, výstup `neexistuje`.
   V opačném případě položíme $\theta := \theta \cup \{X/Z\}$, odstraníme X a Z ze začátků E_1 a E_2 (odstraníme čárky, je-li třeba) a jdeme na krok 2.

2.5.18 Definice resolventy. Máme dvě klauzule K_1 a K_2. Předpokládejme, že K_1 obsahuje literál L_1 a K_2 obsahuje literál L_2 pro něž existuje substituce θ taková, že $\theta(L_1)$ a $\theta(L_2)$ tvoří dvojici komplementárních literálů.

Pak resolventa klauzulí K_1 a K_2 je klauzule K, která je určena všemi literály obsaženými v $\theta(K_1) \setminus \theta(L_1)$ a $\theta(K_2) \setminus \theta(L_2)$; (tj. literály doplněné o obecné kvantifikátory všech proměnných, které se nacházejí v $\theta(K_1) \setminus \theta(L_1)$ a $\theta(K_2) \setminus \theta(L_2)$). □

Neformálně řečeno, resolventu K dostaneme tak, že v „těle“ klauzule necháme všechny literály z v $\theta(K_1) \setminus \theta(L_1)$ a v $\theta(K_2) \setminus \theta(L_2)$ a na začátek doplníme obecné kvantifikátory se všemi proměnnými, které v klauzuli zbyly.

2.5.19 Věta. Jsou dány dvě klauzule K_1 a K_2. Jestliže existuje jejich resolventa K, pak platí: kdykoli jsou pravdivé klauzule K_1 a K_2 v některé interpretaci, pak v této interpretaci je pravdivá i jejich resolventa K. □

Jinými slovy, množiny klauzulí $\{K_1, K_2\}$ a $\{K_1, K_2, K\}$ mají stejné modely.

2.5.20 Rezoluční princip. Je obdobný jako rezoluční princip ve výrokové logice:
Je dána množina klauzulí S. Označme

$$\begin{aligned} R(S) &= S \cup \{K \mid K \text{ je resolventa některých klauzulí z } S\} \\ R^0(S) &= S \\ R^{i+1}(S) &= R(R^i(S)) \quad \text{pro} \quad i \in \mathbb{N} \\ R^\star(S) &= \bigcup \{R^i(S) \mid i \geq 0\}. \end{aligned}$$

Jestliže existuje přirozené číslo n_0 takové, že $R^{n_0}(S) = R^{n_0+1}$. Pak $R^\star(S) = R^{n_0}(S)$.

Věta. Množina klauzulí S je splnitelná právě tehdy, když $R^\star(S)$ neobsahuje prázdnou klauzuli **F**. □